

AperTO - Archivio Istituzionale Open Access dell'Università di Torino

Constitutive and regulative specifications of commitment protocols: A decoupled approach

This is the author's manuscript

Original Citation:

Availability:

This version is available <http://hdl.handle.net/2318/143968> since 2016-06-27T22:08:35Z

Published version:

DOI:10.1145/2438653.2438657

Terms of use:

Open Access

Anyone can freely access the full text of works made available as "Open Access". Works made available under a Creative Commons license can be used according to the terms and conditions of said license. Use of all other works requires consent of the right holder (author or publisher) if not exempted from copyright protection by the applicable law.

(Article begins on next page)

This is the author's final version of the contribution published as:

Matteo Baldoni; Cristina Baroglio; Elisa Marengo; Viviana Patti. Constitutive and regulative specifications of commitment protocols: A decoupled approach. ACM TRANSACTIONS ON INTELLIGENT SYSTEMS AND TECHNOLOGY. 4 (2) pp: 1-25.

DOI: 10.1145/2438653.2438657

The publisher's version is available at:

<http://dl.acm.org/citation.cfm?doid=2438653.2438657>

When citing, please refer to the published version.

Link to this full text:

<http://hdl.handle.net/2318/143968>

Constitutive and Regulative Specifications of Commitment Protocols: a Decoupled Approach

MATTEO BALDONI, CRISTINA BAROGLIO, ELISA MARENGO and VIVIANA PATTI,
Dipartimento di Informatica, Università degli Studi di Torino

Interaction protocols play a fundamental role in multi-agent systems. In this work, after analyzing the trends that are emerging not only from research on multi-agent interaction protocols but also from neighboring fields, like research on workflows and business processes, we propose a novel definition of commitment-based interaction protocols, that is characterized by the decoupling of the *constitutive* and the *regulative* specifications and that explicitly foresees a representation of the latter based on *constraints among commitments*. A clear distinction between the two representations has many advantages, mainly residing in a greater openness of multi-agent systems, and an easier re-use of protocols and of action definitions. A language, named 2CL, for writing regulative specifications is also given together with a designer-oriented graphical notation.

Categories and Subject Descriptors: I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems; I.3.6 [**Methodology and Techniques**]: Interaction techniques; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representations (procedural and rule-based)

General Terms: Theory, Design

Additional Key Words and Phrases: Interaction Protocols, Constitutive and Regulative Specifications, Commitments, Constraints among Commitments

ACM Reference Format:

Baldoni, M., Baroglio, C., Marengo, E., Patti, V. 2011. Constitutive and Regulative Specifications of Commitment Protocols: a Decoupled Approach. ACM Trans. Intell. Syst. Technol. 0, 0, Article 0 (00), 26 pages.
DOI = 10.1145/0000000.0000000 <http://doi.acm.org/10.1145/0000000.0000000>

1. INTRODUCTION

The term “interaction protocol” refers to a pattern of behavior that allows a set of agents to become a multi-agent system (MAS), by engaging expected cooperations. Particularly relevant are *commitment protocols* [Singh 2000; Yolum and Singh 2001; 2002]. Commitments are literals that can hold in the social state of the system, representing the fact that a debtor commits to a creditor to bring about some condition. All agents using a commitment protocol share the semantics of a set of actions which affect the social state. The greatest advantages of commitment protocols, with respect to other approaches to interaction, are that they *do not over-constrain* the specification by imposing unnecessary orderings on the execution of the shared actions, and that by giving a shared (i.e. public and agreed) meaning to the social actions, they allow working on actual knowledge rather than on beliefs about each others’ mental state,

This research has partially been funded by “Regione Piemonte” through the project ICT4LAW.

Author’s address: Matteo Baldoni, Cristina Baroglio, Elisa Marengo and Viviana Patti, Dipartimento di Informatica, Università degli Studi di Torino, c.so Svizzera 185, I-10149 Torino (Italy), {baldoni,baroglio,emarengo,patti}@di.unito.it

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 00 ACM 0000-0003/00/-ART0 \$10.00

DOI 10.1145/0000000.0000000 <http://doi.acm.org/10.1145/0000000.0000000>

thus preserving the agents' autonomy. Nonetheless, commitment protocols *do not yet suit well* those situations where the evolution of the social state is constrained by conventions, laws, preferences or habits, because they do not allow the specification of legal patterns of interaction, although this kind of constraints makes sense in many practical situations, as noticed also in [Singh and Chopra 2010].

In this work, we face this issue by adopting Searle's [1969; 1995] distinction between the *constitutive* and *regulative* specifications of interaction. Roughly speaking, constitutive rules, by identifying certain behaviors as foundational of a certain type of activity, create that activity. They do so by specifying the semantics of actions. Regulative rules, in contrast, contingently constrain a previously constituted activity. In other words, they rule the "flow of activity", by capturing some important characteristics of how things should be carried on in *specific contexts* of interaction [Cherry 1973]. So, for instance, in the well-known Contract-Net Protocol constitutive rules would be used to define the social meaning of the basic actions, like propose, reject, accept. On top of these it is possible to specify regulative rules, like the fact that proposals must follow a call-for-proposals, and that each proposal must be acknowledged with an accept or a reject. Notice that on top of the same basic actions it would be possible to specify other regulative rules, that may, for instance, correspond to variations of the Contract-Net Protocol, fitting different contexts and needs. In these contexts, a proposal could, for instance, be considered even though no call was issued.

Given a domain, it is not easy to distinguish, in general, what is constitutive from what is regulative; this depends on the specific problem to be solved and on the abstraction level that is required (analogously to what happens with the implementation of abstract data types) but once this has been defined, there is the need to represent both levels. A clear separation in such representation of the constitutive from the regulative specification would bring many advantages, mostly as direct effects of the obtained *modularity: easier re-use* of actions in different contexts, *easier customization* on the protocol, *easier composition* of protocols. As a consequence, MAS would gain greater *openness, interoperability*, and *modularity of design*. Interoperability would be better supported because it would be possible to verify it with respect to specific aspects (e.g. at the level of actions [Chopra and Singh 2008; Chopra 2009; Chopra and Singh 2009b] or at the level of regulation rules [Baldoni et al. 2011]). Protocols would be more open in the sense that their modularity would allow designers to easily adapt them to different needs (see Section 4). Agents could also check individually (against the protocol specification) if they have actions that *match* with the constitutive rules independently from the regulative specification.

In the light of the distinction between constitutive and regulative rules, this work analyzes alternative proposals for the specification of interaction protocols based on commitments (Section 2), showing that, despite the fact that it is possible to recognize various attempts to capture both specifications, these proposals still miss the clear distinction postulated in [Searle 1969; Cherry 1973] and described above. We show that none allows the specification of both parts (1) *in a decoupled way*, (2) *by means of first-class languages*, (3) *which allow flexible representations* – either one of the two specifications is disregarded or it is too strict or the two representations are to some extent mixed. Section 3, then, proposes a model for commitment-based interaction protocols that separates the constitutive and the regulative parts. In particular, we extend commitment-based protocols, as proposed in [Chopra and Singh 2008; Chopra 2009], by adding to the constitutive specification of actions, an explicit regulative specification, given as a set of *constraints among commitments*. For representing the regulative specification we propose the language 2CL, that allows the specification of different kinds of constraints. The language is characterized by a graphical notation, aimed at supporting the designer in the specification of interaction protocols. It inherits from

[Pesic and van der Aalst 2006; Montali et al. 2010] but it is very different from it in its basic principles. Indeed, 2CL builds on *commitments*, thus ruling the evolution of the social state, and not on events (actions). Section 4 shows how it is easy to tailor an interaction protocol, expressed in 2CL, to different contexts of use, by modifying the regulative specifications only. As a running example we chose the Contract Net Protocol (CNP) [Foundation for Intelligent Physical Agents 2002] but in Section 3.2 we also report a representation of a fragment of the Robert’s Rules of Order, which is a more complex interaction protocol and, as such, helps at showing the usefulness of a wider range of 2CL operators. In Section 5 we compare commitments and constraints for what concerns their use in the specification of regulative rules, with a particular attention to the meaning of *violation*. In Section 6 we finish the comparison with the models in Section 2, showing that our proposal includes the others as a special case or overcomes their limits, and report the future directions of research.

2. ACTIONS AND PROTOCOLS: CONSTITUTIVE AND REGULATIVE SPECIFICATIONS

Let us consider commitment-based protocols. Commitments are directed from a debtor to a creditor. The notation $C(x, y, r, p)$ denotes that agent x commits to an agent y to bring about consequent condition p when the antecedent condition r holds. In the special case when r equals *true*, we use the short notation $C(x, y, p)$. Agents share a social state that contains commitments and other literals that are relevant to their interaction. Every agent can affect the social state by executing actions, whose definition is given in terms of operations onto the social state (e.g. adding a new commitment, releasing another agent from some commitment, satisfying a commitment, etc. see [Yolum and Singh 2002]). On the other hand, the behavior of agents is affected by the commitments because the underlying assumption is that agents respect the commitments they have taken. In other words, commitments have a *regulative* nature.

Usually a commitment protocol is made of a set of actions, whose semantics is known to (and agreed upon by) all of the participants [Yolum and Singh 2002; 2001; Chopra 2009]. There are many definitions for actions in the literature. In *UML* and in the literature about workflows, actions are atomic executions. They are considered to take zero time, and cannot be interrupted, while activities represent more complex behaviors, that may run for a long time, and may be interrupted by events. Most of works on agents adopt, instead, a *precondition-effect* view of actions. *Preconditions* can be of two kinds: preconditions to the action execution, and preconditions to some effect. The former are literals that must hold in the social state to make the action executable, the latter are additional conditions that, when holding, allow the production of the specific effect that they control. For instance, in order to pay by credit card it is necessary to own a credit card (precondition to the action). If a credit card owner uses it for paying, the payment will be done only if the card is valid (conditional effect). For example, in [Chopra and Singh 2008; Chopra 2009] actions have no preconditions of any kind, in [Chopra and Singh 2006; Kafali and Yolum 2009] actions have both preconditions to the executability and conditional effects, while Winikoff *et al.* [2005] use only preconditions to the execution of actions. Whatever representation is chosen, the protocol actions are specified by means of *constitutive rules*.

Commitments have a “regulative” nature in the sense that agents that take a commitment are bound to make the condition in the commitment become true, however, it is often necessary to specify *patterns of interaction* which require a degree of expressiveness that, as we will show, commitments alone do not have. Contrarily to constitutive rules, which define new forms of behavior, these patterns regulate antecedently existing forms of behavior [Searle 1995]. Therefore, such rules regulate the social reality, defined by the constitutive rules. For this reason we refer to them as the “regulative specification of the protocol”. For example, a purchase protocol may state that

the payment must occur first in order for the shipment to proceed. By means of this rule, an agent whose goal is having the goods shipped, would conclude that, in order to achieve its goal when interacting with partners that comply to this protocol, it must have the payment done first. A regulative specification may, therefore, be viewed as encoding a *policy*. Let us, therefore, overview the most relevant works in the context of *commitment-based interaction protocols*, with a particular focus on the way they specify *regulative rules*, in order to compare and discuss the proposed models, which are graphically summarized in Fig. 1 and Fig. 2.

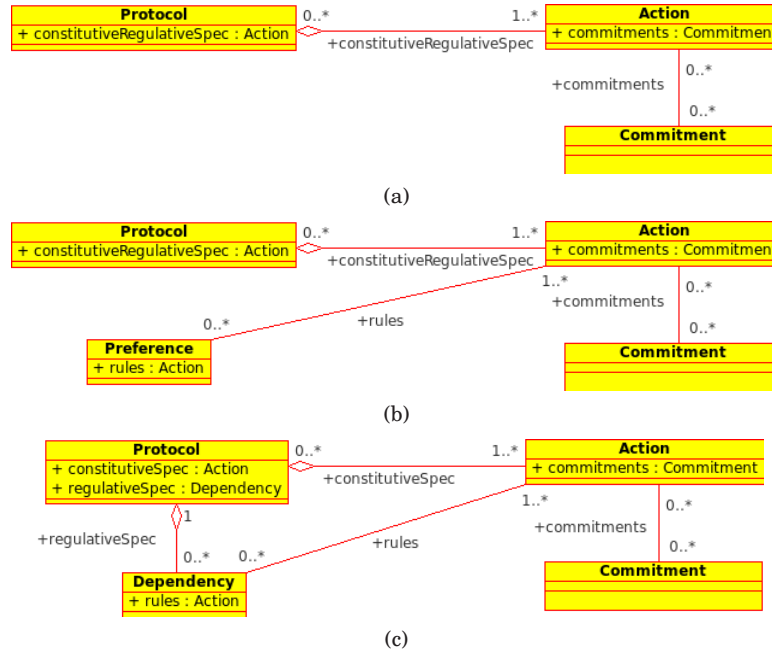


Fig. 1. (a) Chopra and Singh's implementation model: regulative specifications based on actions; (b) Mallya and Singh's model: adding preferences on actions; (c) Singh's dependencies among events.

Commitment-based protocols

Chopra and Singh [2008] introduce the distinction between constitutive and regulative specifications in the definition of commitment protocols but study in depth only the constitutive part. Each agent is publicly described by the *effects* of the actions it can execute (see above), which make the *constitutive specification* of the agent. Such specifications allow agents to agree on the meaning of their actions. This approach can be found also in [Yolum and Singh 2002; 2001; Winikoff et al. 2005; Chopra and Singh 2006; Giordano et al. 2007; Chopra 2009; El-Menshawey et al. 2010; 2011]. All these approaches do not account for regulative specifications, besides the one given by commitments. As a matter of fact, in this context, the flow can only be controlled by the *preconditions to the (non-)executability of the actions*, see Fig. 1(a). So, in order to impose that sending goods should follow payment, the action *send-goods* should have as a precondition a literal that is made true as an effect of the action *pay*. This solution has two limits. The first, and most important one, is that it does not abide by the meaning of “regulative” rule, i.e. to regulate antecedently existing forms of behavior. Preconditions are, in fact, part of the constitutive specification of the actions because

they help defining the rules of the game and to specify the possibility of action [Conte 1983; Searle 1995; La Torre 2010; Grossi 2007]. Moreover, we can say that their use forces the *regimentation* of the regulation [Jones and Sergot 1994] because it is not possible to execute actions leading to violations. The adoption of regimentation, rather than enforcement, is, in our opinion, a choice that should be left up to the designer of the system and not be imposed by the specification framework [Grossi et al. 2007].

Preferences and dependencies

Mallya and Singh [2006], Fig. 1(b), propose to order the possible executions according to a set of preferences that take into account the policies of the various parties. No execution is strictly forbidden but a preference criterion is specified. The constitutive specification is given in terms of commitments but the preference rules are given in terms of actions. Preferences do not precisely correspond to regulative rules because they specify selection policies rather than defining what is allowed (or forbidden). For this reason they can be interpreted as guidelines. Nevertheless, giving them in terms of actions makes the specification less flexible and less easily adaptable or open. The same limits, Fig. 1(c), can be ascribed to the work to which [Mallya and Singh 2006] is inspired, i.e. [Singh 2003], although in this work it is possible to recognize the introduction of a regulative specification, based on the *before* relation applied to events.

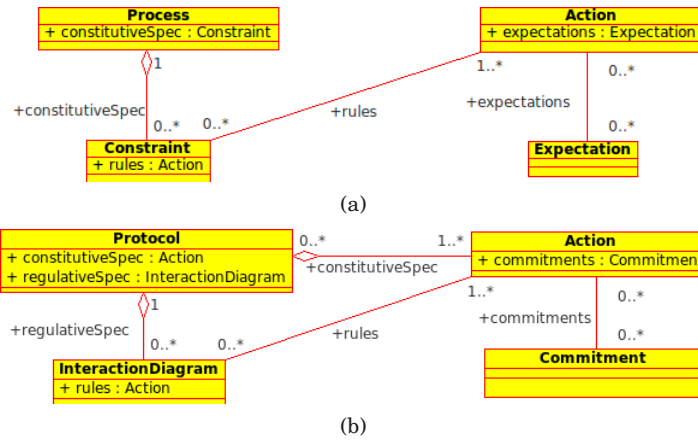


Fig. 2. (a) ConDec model: constitutive specification given by means of constraints on actions, an extension supplies an expectation-based semantics for actions; (b) Fornara and Colombetti's model: regulative specification given by interaction diagrams defined on actions.

ConDec

Pesic and van der Aalst [2006], Fig. 2(a), propose to use the declarative language ConDec for representing business processes. The approach does not build on commitments nor is set in the agents framework. In [Montali 2010; Chesani et al. 2009; Montali et al. 2010], the authors use this approach to specify interaction protocols and service choreographies. To this aim, they integrate ConDec with SCIFF thus giving a semantics to actions that is based on expectations [Alberti et al. 2004]. A ConDec process is defined by means of LTL temporal constraints. The operational semantics of a ConDec process is the Büchi automaton that is obtained by translating the constraints, applying well-known algorithms from the field of Model Checking [Clarke and Peled 2001] and refined in [Pesic et al. 2010]. The aim is to use the declarative approach to simplify the

design of processes. The nature of the specification is *constitutive* because it defines a behavior rather than regulating an antecedently existing reality. The constitutive specification is given at two levels: the level of constraints that builds upon the level of the actions. Even in the case one uses the above model not just to design processes but with a regulative intent, other problems emerge, due to the fact that constraints are defined over actions (events), see Fig. 2(a). This, in our opinion, clashes with the openness of MAS. Let us explain our view with an example. Let us suppose that payment should occur before sending the goods, and that the protocol foresees the actions *pay-by-credit-card* and *send-goods*. Then, it will specify that *pay-by-credit-card* must occur before *send-goods*. Now, if a client arrives which can pay cash, it will not be in condition to take part to the interaction unless the specification is changed by adding a rule that says that paying cash should occur before sending the goods. This should be done even though the action has the same semantics of *pay-by-credit-card* in terms of commitments. The need of modifying the specification (even in the case when actions have the same semantics!), gives *an undesired rigidity to the protocol*. Problems arise also in the case an agent can execute a sequence of actions which *altogether* implement one of those foreseen by the protocol. The fact that the specification is given in terms of *actions* implies that if an action name changes, one needs to revise the process definition as well. It is also easy to make mistakes by forgetting to update the specification when a new action is changed or when its semantics is changed.

Fornara and Colombetti

Fornara [2003] and Fornara and Colombetti [2004a; 2004b], Fig. 2(b), define a commitment-based semantics for the speech acts of agent communication languages, like FIPA, and then use *interaction diagrams* to define the regulative specification. In this proposal, the social actions are represented by the speech acts and the constitutive specification is given in terms of commitments. The choice of relying on interaction diagrams is, however, very strong because it forces the ordering of action executions, loosing, in our opinion, the flexibility aimed at by the adoption of commitments.

Summary

The distinction between a regulative and a constitutive specification is surely interesting but the current proposals show some limits in the realization of this model, each with its pros and cons. Fornara and Colombetti's proposal is too rigid: the use of interaction diagrams conflicts with the desirable flexibility of commitments. In this respect, ConDec's use of constraints is better: the declarative approach that is proposed is aligned with the declarative nature of commitments. The problem is that constraints are used with a constitutive and not with a regulative aim. In other words, the ConDec approach uses the constraints to define a process with its execution traces. Moreover, they are defined in terms of performing actions rather than of bringing about conditions. Chopra and Singh [2008] propose an implementation where the regulative specification is given on top of actions: while commitments are given on conditions and not on the actions that should bring them about, constraints are posed on the action execution, with the result that modularity is not obtained. The same holds for [Winikoff et al. 2005; Giordano et al. 2007; Yolum and Singh 2002; 2001; Singh 2003; El-Menshawey et al. 2010; 2011]. Our proposal (see Figure 3) aims at overcoming the listed limits. We extend commitment-based protocols by adding the possibility of defining expressive regulative specifications, that are not limited to commitments but that account for patterns of interaction. To this aim we propose the use of a declarative language, named 2CL, which allows the definition of constraints on the evolution of the social state. Such specifications preserve the flexibility that is typical of commitment protocols because, in general, they do not force agents to execute given paths

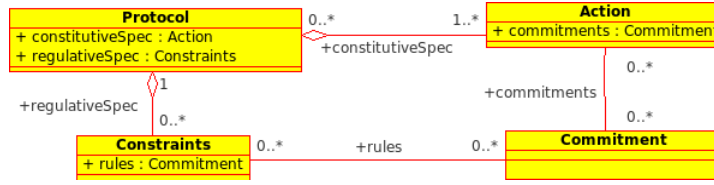


Fig. 3. Our proposal: decoupling between constitutive (actions) and regulative (constraints) specifications.

but rather let them free to choose their courses of actions, as long as they respect the rules. 2CL constraints relate commitments and not actions/events. This modularity facilitates the design of interaction protocols because it allows for a separate specification of the actions and of the regulative part, once the vocabulary of terms (facts, commitments) that can appear in the social state is defined. It also facilitates the reuse of previously defined actions or regulations as long as the domain of discourse does not change. Regulations, in fact, are no more hidden inside actions, and actions are no more over-specified due to the need of including regulations in their preconditions. This is, for instance, useful in the case when a designer must tailor an interaction protocol to particular needs, a case that will be discussed in the following. Another characteristic of the proposed language is that it is a general tool, that leaves the designer free to choose how to implement the specified regulations (e.g. by enforcement or by regimentation) rather than forcing him/her in any direction.

3. COMMITMENT PROTOCOLS: A DECOUPLED APPROACH

In this work, we propose an approach to the definition of commitment-based interaction protocols which extends the classic approach, based on the *constitutive* specification of the meaning of the social actions, by adding a *regulative* specification, which specifies the legal evolutions of the social state. Both are defined based on *commitments*.

Definition 3.1 (Interaction protocol). An interaction protocol P is a tuple $\langle Ro, F, A, C \rangle$, where Ro is a set of roles, identifying the interacting parties, F is a set of literals (including commitments) that can occur in the social state, A is a set of actions, and C is a set of constraints.

The set of social actions A , defined on F and on Ro , forms the *constitutive specification* of the protocol, while the set of constraints C , defined on F and on Ro too, forms the *regulative specification* of the protocol.

In the following, we use as a running example the well-known Contract Net Protocol (CNP for short) [Foundation for Intelligent Physical Agents 2002]. Briefly, CNP includes two roles, the initiator (i in the following) and the participant (p). The initiator calls for proposals. The participant may send a proposal or refuse to do it. When a proposal is received, the initiator may either reject or accept it. Notice that, for the sake of simplicity, we do not model the exchange of information concerning the proposal itself but only the interaction concerning the task assignment and solution. We report the CNP as represented according to our proposal, by giving its constitutive specification followed by its regulative specification.

Constitutive specification

For what concerns the *constitutive specification* of an action, we adopt the same representation as in [Chopra 2009], and define its meaning in terms of how it affects the social state by adding or removing literals or by performing operations on the commit-

ments (create, release, delete, etc., see [Singh 1999; Yolum and Singh 2001]). The constitutive specification follows the grammar below, where the *means* construct amounts to a *counts-as* relation [Searle 1995; Jones and Sergot 1996]:

$$\begin{aligned} A &\rightarrow (Action \textbf{ means } Operation)^+ \\ Action &\rightarrow protocolAction([paramList]) \\ Operation &\rightarrow Op(commitment) \mid fact \mid Operation \wedge Operation \\ Op &\rightarrow CREATE \mid DELETE \mid RELEASE \mid DELEGATE \mid ASSIGN \mid \dots \end{aligned}$$

where *protocolAction* is the name of an interactive action of the protocol (not necessarily having the form of a communication or a message but possibly amounting to any observable action); *paramList* denotes the possible parameter list of the action; *Op* is one of the operations on commitments; *commitment* is a commitment of form $C(x, y, r, p)$, as specified in Section 2 (see also [Chopra 2009, page 49]), where x and y are roles in R_o and r and p are formulas in disjunctive normal form of propositional literals in F ; and *fact* is a positive or negative proposition that does not concern commitments and which contributes to the social state (they are the conditions that are brought about – when necessary they are enriched by two parameters: the actor and the recipient).

As an example, let us consider the actions foreseen by CNP as expressed according to the above grammar:

- (a) $send_cfp(i, p) \textbf{ means } CREATE(C(i, p, assigned_task(i, p)))$
- (b) $send_proposal(p, i) \textbf{ means } CREATE(C(p, i, solved_task(p, i)))$
- (c) $send_refusal(p, i) \textbf{ means } refused_task(p, i) \wedge RELEASE(C(i, p, assigned_task(i, p)))$
- (d) $send_accept(i, p) \textbf{ means } assigned_task(i, p)$
- (e) $send_reject(i, p) \textbf{ means } rejected_proposal(i, p) \wedge$
 $DELETE(C(i, p, assigned_task(i, p))) \wedge RELEASE(C(p, i, solved_task(p, i)))$
- (f) $send_done(p, i) \textbf{ means } solved_task(p, i)$
- (g) $send_failure(p, i) \textbf{ means } failed(p, i) \wedge DELETE(C(p, i, solved_task(p, i)))$

Since such definitions are quite straightforward, we get into the details of just a couple of them. The effect of action $send_cfp$ is to create the commitment $C(i, p, assigned_task(i, p))$. Intuitively, this commitment states the resolution of the initiator to assign a task to a participant because it needs someone to solve it. Indeed, during the execution the participant may refuse to solve the task or the initiator may reject a proposal because, for example, it is not convenient. The action $send_refusal(p, i)$, by which the participant refuses to solve a task, instead, has as effect the action $RELEASE(C(i, p, assigned_task(i, p)))$, by which the participant releases the initiator from the commitment of assigning a task to it, and the fact $refused_task(p, i)$, whose meaning is clear.

Regulative specification

In order to represent the *regulative specification*, we propose a declarative language named 2CL (the acronym stands for “Constraints among Commitments Language”). The language has also a graphical notation which is inspired by ConDec [Pesic and van der Aalst 2006] and by [Baldoni et al. 2011]. By means of it, the designer can express many kinds of constraints describing the legal evolutions of the social state. As underlined in [Pesic and van der Aalst 2006; Baldoni et al. 2011; Montali 2010], constraint-based declarative representations provide abstractions which allow to explicitly capture what is mandatory and what is forbidden, without the need to express the set of possible executions extensionally. For this reason, models remain compact improving flexibility: they specify what is desired and undesired, leaving all that remains unconstrained. This is an advantage with respect to procedural approaches,

characterized by a prescriptive nature which requires the specification of *all* the allowed evolutions. It also accommodates naturally to the commitment-based approach, where a central issue is the respect of the agents' autonomy.

2CL specifications follow the grammar:

$$\begin{aligned} C &\rightarrow (Disj \text{ op } Disj)^* \\ Disj &\rightarrow Conj \text{ OR } Disj \mid Conj \text{ XOR } Disj \mid Conj \\ Conj &\rightarrow literal \text{ AND } Conj \mid literal \end{aligned}$$

C , see Def. 3.1, is a set of constraints of the form $A \text{ op } B$, where A and B are formulas of literals and op is one of the operators described below; *literal* can be either a commitment or a positive or negative proposition (where negation means that a certain literal does not hold in the social state).

The operators of 2CL are reported in Table I, together with their meaning. For each relation, there are two types of constraint: *base* and *persistence*. Constraints of type *base* express relations between the literals, saying what should become true in the social state and when. Constraints of type *persistence* capture the fact that a condition of interest should hold in all the traversed states, until another condition of interest becomes true. The graphical representation of the operators follows the lines of [Pesic and van der Aalst 2006; Baldoni et al. 2011], therefore it respects a few conventions: single lines represent base operators, while triple lines represent persistence operators; the position of the dot intuitively denotes the occurrence of which condition activates the relation expressed by the constraint; arrows, when used, and their relative positions with respect to dots, denote the qualitative temporal nature of the constraint; negated operators are crossed by a line.

The proposed operators capture those kinds of constraints that the analysis of protocols like Contract Net, Net Bill [Baldoni et al. 2011] and Robert's Rules of Order [Baldoni et al. 2010] (see Section 3.2) suggest being the most useful for the regulative specification of interaction protocols. Let us consider a few significant ones. *Correlation* ($\bullet\rightarrow$) is one of those operators expressing the fact that two conditions occur in a same execution. Specifically, the dot is on the left-hand side and this represents the fact that the constraint is triggered when the condition on that side becomes true. In this case, the execution is requested to contain some state in which also the condition on the right-hand side becomes true. The absence of arrows denotes that there is no requested temporal relation between this state and the former one. Therefore, in order for the execution to satisfy the constraint, the condition on the right-hand side may become true in future or have been true in the past, or even in the same state as the one in which the left-hand side became true. The *before* ($\rightarrow\bullet$) operator denotes, instead, a temporal relation: in fact, an arrow appears at its rightmost end. Also the dot is at that end of the operator. This means that the constraint is triggered when, during an execution, the condition on the right-hand side of the constraint becomes true. Due to the temporal relation expressed by the arrow, the condition on the left-hand side must hold in some state of the same execution that precedes the former. The operator $\bullet\rightarrow\bullet$ has two dots and one arrow: it captures a form of *causality* between the condition on the left-hand side and the one on the right-hand side, of the kind foreseen in [Lamport 1978]. *Negative constraints* (e.g. \nrightarrow) denote evolutions of the social state that should be avoided. The meaning of the one used as an example is that when the condition on the left-hand side holds in some state of an execution, the other condition should not hold in any of its states. *Persistence constraints* denote conditions that hold along time. For instance, *persistence response* ($\bullet\Rightarrow$) denotes the fact that the condition on the left-hand side should hold along all the states traversed by the execution from the state in which it becomes true through the state in which the condition on the right-hand side becomes true.

The proposal corrects and extends some of the representation models discussed in Section 2. Specifically, we can model the regulative rules that Chopra and Singh [2006], as well as Giordano *et al.* [2007], Yolum and Singh [2001], and Winikoff *et al.* [2005], (which follow the same principles) represent by means of preconditions by introducing for each action a literal that is univocally associated to it, as an effect of the action, and, then, to define constraints (typically of kind *premise*, \bowtie) among these literals, without forcing the regimentation imposed when directly working on action preconditions. The use of constraints of kind *premise* also allows the designer to specify *strict sequences* of action, as in [Fornara and Colombetti 2003; 2004a; Fornara 2003]. Last but not the least, the proposal overcomes the limits of [Pesic and van der Aalst 2006; Montali 2010; Chesani et al. 2009; Montali et al. 2010] because it introduces a regulative specification that specifies the legal evolutions of the social state and does it at the level of commitments and not at the level of actions/events.

The operators can be used to write declarative constraints, which can also be given a *graphical format*, as in Figures 4 and 6. Here a rectangle denotes a conjunction of literals (commitments or facts) that can be in the social state; a 2CL operator connects different rectangles, hence defining a relation between the contents of the social state at different times. The related states do not need to be immediately subsequent. The set of all the relations specified by means of 2CL operators specifies how the social state is expected to evolve when the protocol is respected. The graphical notation also supplies elements for denoting *conjunctions* (circles), *disjunctions* (diamonds) and *exclusive or's* (bordered diamonds, e.g. see Figure 4).

Going back to the Contract Net Protocol example, the regulative rules can be specified as follows:

- c1: $C(i, p, assigned_task(i, p)) \bullet \rightarrow C(p, i, solved_task(p, i)) \text{ XOR } refused_task(p, i)$
- c2: $C(p, i, solved_task(p, i)) \bullet \rightarrow rejected_proposal(i, p) \text{ XOR } assigned_task(i, p)$
- c3: $assigned_task(i, p) \bullet \rightarrow solved_task(p, i) \text{ XOR } failed(p, i)$

Fig. 4 reports them as a graph, for the sake of readability we omitted parameters of literals in the figures. The initiator declares its intention to assign a task (node $n1$, $C(i,$

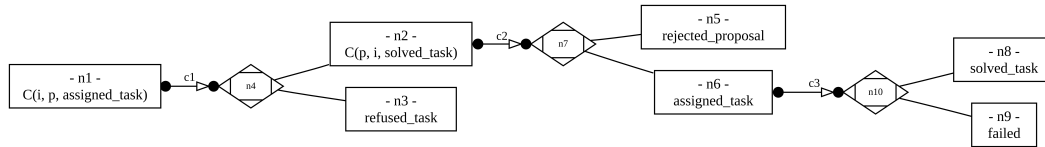


Fig. 4. Regulative specification of the Contract Net Protocol.

$p, assigned_task(i, p))$). If this happens, afterwards the participant takes its decision and alternatively refuses or states its intention to solve the task. This is represented by the fact that the node $n1$ is connected to the nodes $n2$ ($C(p, i, solved_task(p, i))$), and $n3$ ($refused_task(p, i)$): $n2$ and $n3$ are alternative evolutions of the social state after $n1$. The connector $n4$ denotes the *exclusive or* of the two. The *cause* arrow $\bullet \rightarrow$ represents the fact that when the initiator must assign a task, the participant necessarily has to either refuse the task or take the commitment to solve it. It is not obliged to take one of these two ways at the next step of its execution but sooner or later it is expected to do it. The specification foresees that the participant cannot take the initiative of proposing to solve a task (or of refusing to do something) if the initiator has not declared that there is a task to solve. This is the intuitive meaning of the circles at the two sides of the arrow $c1$.

Table I. 2CL operators and their meaning.

<i>Relation</i>	<i>Type</i>	<i>Operator</i>	<i>Meaning</i>
Correlation	base (pos.)	$A \bullet - B$	In an execution where A occurs, also B must occur but there is no temporal relation between the two.
	base (neg.)	$A \nrightarrow B$	If A occurs in some execution, B must not occur.
	persistence (pos.)	$A \bullet = B$	B must occur in every state where A occurs.
	persistence (neg.)	$A \nrightarrow = B$	When A occurs, B cannot occur in the same state.
Co-existence	base (pos.)	$A \bullet \bullet B$	It captures the mutual correlation $A \bullet - B$ and $B \bullet - A$.
	base (neg.)	$A \nrightarrow \bullet B$	This captures the mutual exclusion of A and B : both $A \nrightarrow B$ and $B \nrightarrow A$ hold.
	persistence (pos.)	$A \bullet = B$	This expresses the mutual persistent correlation between A and B : $A \bullet = B$ and $B \bullet = A$.
	persistence (neg.)	$A \nrightarrow = B$	This captures the mutual persistent exclusion of A and B : $A \nrightarrow B$ and $B \nrightarrow A$.
Response	base (pos.)	$A \bullet \rightarrow B$	If A occurs, B must hold at least once afterwards (or in the same state). It does not matter if B already held before A .
	base (neg.)	$A \nrightarrow \rightarrow B$	If A holds, B cannot hold in the same state or after.
	persistence (pos.)	$A \bullet \Rightarrow B$	A persists until B becomes true.
	persistence (neg.)	$A \nrightarrow \Rightarrow B$	If A occurs, it does not persist until B .
Before	base (pos.)	$A \rightarrow \bullet B$	B cannot hold until A becomes true. Afterwards, it is not necessary that B becomes true.
	base (neg.)	$A \nrightarrow \bullet B$	In case B becomes true, A cannot hold beforehand.
	persistence (pos.)	$A \Rightarrow \bullet B$	B does not hold until A becomes true, and afterwards A holds until B becomes true.
	persistence (neg.)	$A \nrightarrow \bullet B$	In case B becomes true, A cannot hold beforehand.
Cause	base (pos.)	$A \bullet \rightarrow \bullet B$	It is the conjunction of the base <i>response</i> and base <i>before</i> relations: $A \bullet \rightarrow B$ and $A \rightarrow \bullet B$.
	base (neg.)	$A \nrightarrow \bullet \rightarrow B$	It is the conjunction of the base <i>response</i> and base <i>before</i> negative relations: $A \nrightarrow \rightarrow B$ and $A \nrightarrow \bullet B$.
	persistence (pos.)	$A \bullet \Rightarrow \bullet B$	It is the conjunction of the persistent <i>response</i> and persistent <i>before</i> relations: $A \bullet \Rightarrow B$ and $A \Rightarrow \bullet B$.
	persistence (neg.)	$A \nrightarrow \Rightarrow \bullet B$	It is the conjunction of the persistent <i>response</i> and persistent <i>before</i> negative relations: $A \nrightarrow \Rightarrow B$ and $A \nrightarrow \bullet B$.
Premise	base (pos.)	$A \bowtie - B$	It is a stronger temporal relation concerning <i>subsequent</i> states, stating that A must hold in the state immediately preceding one state in which B holds.
	base (neg.)	$A \nrightarrow \bowtie B$	It states that A must never hold in a state that immediately precedes one where B holds.

Notice that we have not mentioned which actions should be executed to change the social state. In this case, in fact, any action, whose effect is compatible with the schema of evolution of the social state reported above is feasible (in the same way as it is not necessary, in commitment protocols, to say which action to take to satisfy a commitment). Notice, however, that the proposed approach can also capture the case in which it is necessary to achieve an effect by executing a specific action: this can be done by adding to the constitutive rule, that defines the action at issue, a literal that is univocally associated to it, as one of its effects, and, then, to define a proper constraint that uses this literal.

Moreover, the transition from one state to one of its next (with respect to the description given by regulative specification) states may require the application of many actions (not necessarily one). The regulative specification does *not* give any *procedure* for achieving the social state change, that it captures. In fact, constraints that define the legal evolutions of the social state are independent from the actions that are used by the agents. Both, however, are specified on top of the literals in the social state.

If the interaction continues because the participant has proposed to solve the task, the initiator must either reject the proposal or accept it and assign the task to the participant, which, in this case, is expected to execute it and to give back to the initiator an answer (the solution or the information that it has failed). The arrows in the graph between nodes n_2 and the alternative between n_5 and n_6 , on a side, and between n_6 and the alternative between n_8 and n_9 are again causality operators.

3.1. Semantical considerations

An interaction protocol specifies in a concise way a set of legal interactions. The interaction of a set of agents will be compliant to the protocol when all the commitments they have towards the others, and that are objectively inferrable from their observable behavior, are satisfied (as usual in the social approach), and the overall execution respects the constraints that make the regulative specification. For what concerns the formal semantics of the commitment-based protocols that were discussed in Section 2, some of them propose an operational semantics that relies on commitment machines to specify and execute protocols [Yolum and Singh 2002; 2001; Winikoff et al. 2005; Chopra and Singh 2006; Chopra 2009], some others, like [Fornara and Colombetti 2004a; 2004b], use interaction diagrams, operationally specifying commitments as an abstract data type, and analysing the commitment's life cycle as a trajectory in a suitable space. Other approaches rely on temporal logics to give a formal semantics to commitments and to the protocols defined upon them. Among these, [Giordano et al. 2007] uses DLTL, while [Bentahar et al. 2009; El-Menshawry et al. 2010; 2011] adopt extensions of CTL*. Given the specification of a set of social actions, all these approaches allow the inference of those executions, which are legal with respect to the protocol. Intuitively, the addition of a regulative specification by means of 2CL constraints restricts the set of acceptable executions.

In particular, commitment machines [Yolum and Singh 2002] (later refined in [Winikoff et al. 2005]) specify the possible states of an execution, the actions that are used for doing the transitions, and the possible final states of the protocol. The meaning associated with each state specifies which commitments are in force in that state, and the meaning associated with each action defines how the commitments are affected by the action, leading to a state change. Intuitively, commitment machines allow the formalization of legal executions by taking into account only the *constitutive* specification of the social actions.

2CL allows the expression of temporal constraints on execution paths. Since commitment machines express legal execution paths, temporal constraints can be used to restrict this set. A natural choice for formalizing such constraints is *Linear Temporal*

Table II. 2CL operators and their semantics in LTL.

Relation	Type	Positive	LTL meaning	Negative	LTL meaning
Correlation	base	$A \bullet \dashv B$	$\Diamond A \supset \Diamond B$	$A \not\bullet \dashv B$	$\Diamond A \supset \neg \Diamond B$
	persistence	$A \dashv B$	$\Box(A \supset (A \wedge B))$	$A \not\bullet \dashv B$	$\Box(A \supset \neg(A \wedge B))$
Co-existence	base	$A \bullet \bullet B$	$A \bullet \dashv B \wedge B \bullet \dashv A$	$A \not\bullet \bullet B$	$A \not\bullet \dashv B \wedge B \not\bullet \dashv A$
	persistence	$A \dashv \bullet B$	$A \dashv B \wedge B \dashv A$	$A \not\bullet \bullet B$	$A \not\bullet \dashv B \wedge B \not\bullet \dashv A$
Response	base	$A \bullet \bullet B$	$\Box(A \supset \Diamond B)$	$A \not\bullet \bullet B$	$\Box(A \supset \neg \Diamond B)$
	persistence	$A \dashv \bullet B$	$\Box(A \supset (\Diamond B \wedge (A \cup B)))$	$A \not\bullet \bullet B$	$\Box(A \supset \neg(A \wedge B))$
Before	base	$A \rightarrow B$	$\neg B \cup A$	$A \not\rightarrow B$	$\Box(\Diamond B \supset \neg A)$
	persistence	$A \dashv \bullet B$	$\neg B \cup (A \cup B)$	$A \not\rightarrow B$	$\Box(\Diamond B \supset \neg A)$
Cause	base	$A \bullet \bullet B$	$A \bullet \dashv B \wedge A \rightarrow B$	$A \not\bullet \bullet B$	$A \not\bullet \dashv B \wedge A \not\rightarrow B$
	persistence	$A \dashv \bullet B$	$A \dashv B \wedge A \dashv \bullet B$	$A \not\bullet \bullet B$	$A \not\bullet \dashv B \wedge A \not\rightarrow B$
Premise	base	$A \dashv \bullet B$	$\Box(\Diamond B \supset A)$	$A \not\bullet \bullet B$	$\Box(\Diamond B \supset \neg A)$

Logic (LTL) [Emerson 1990], which includes temporal operators like next-time ($\bigcirc\varphi$, the formula φ holds in the immediately following state of the run), eventually ($\Diamond\varphi$, φ is guaranteed to eventually become true), always ($\Box\varphi$, the formula φ remains invariably true throughout a run), weak until ($\alpha\cup\beta$, the formula α remains true until β becomes true). This kind of logic allows the identification of those executions which satisfy the constraints of interest. We, therefore, introduce a LTL semantics to the operators of 2CL (Table II), formalizing the intuitive semantics given in Table I. In particular, the *premise* operator is the only one to express a temporal relation over subsequent states; for this reason the LTL formula is expressed by means of the \bigcirc LTL operator. *Co-existence* and *cause*, instead, derive respectively from correlation, and from before and response. Notice that the negated operators semantics (column 6) not always corresponds to the negation of the semantics of the corresponding positive operator (column 4). This is due to the intention of capturing the intuitive meaning of negations. We show this need by means of a couple of examples. For what concerns correlation, the negation of the formula in column 4, which is $\Diamond A \wedge \neg \Diamond B$, is too strong because it says that A must hold sooner or later while B cannot hold. What we mean by negated co-existence, instead, is that *if A becomes true then B must not occur in the execution*. For completeness, the semantics of negated correlation is not equivalent to the semantics of $A \bullet \dashv \neg B$. For what concerns *premise*, by negating the semantics in column 4 we obtain $\Diamond(\bigcirc B \wedge \neg A)$ which says that B occurs in some state and A does not occur in the previous state. Instead, the intended meaning is that A does not have to hold in the states that precede those in which B holds (but B does not necessarily have to hold). Analogous considerations can be drawn for the other operators.

A *legal execution* of a commitment-based protocol, enriched by means of 2CL regulative specifications, is an execution that is accepted by the commitment machine built upon the constitutive specification of the protocol, and that, when interpreted as a linear temporal model, satisfies the LTL formulas corresponding to the regulative specification of the protocol. Based on this characterization, it is possible to provide mechanisms for verifying that an agent is behaving in respectance to the protocol. When this does not happen, we can say that a *violation* has occurred. So, if in a commitment-based protocol, made only of the constitutive specification of a set of actions, violations are detected only when in a final state a commitment remains unsatisfied, in our proposal we also detect violations during the execution, due to fact that a constraint is not respected. For what concerns the verification that a linear temporal model, corresponding to an execution, respects the 2CL constraints, one could envisage the use of model checking techniques, along the lines of what was done in a previous work [Baldoni et al. 2011].

3.2. Robert's Rules of Order

Let us, now, introduce a more complex and actually used interaction protocol (see also [Pitt et al. 2006]), in order to test the effectiveness of the proposed specification: the *Robert's Rules of Order* (RONR for short [Robert et al. 2000]). RONR is a well-known regulation of the behavior to be followed by a democratic deliberative assembly, like the Parliament, to discuss and decide about issues called *motions*. This protocol is interesting because not only it specifies the social actions but it also encompasses rules that govern the behavior of the assembly. Briefly, it states that before voting a motion, everybody who wishes to speak must have the possibility of speaking; in order to speak it is necessary to have the floor, and it is not possible to speak while someone else is speaking. No limit is imposed on the duration of the discussion, and it is up to the chair to decide to whom assigning the floor. In the following we denote by c the chair of the assembly and by p a generic participant, P is the set of all the members of the assembly, and m is the discussed motion. This protocol will help at showing the usefulness of a wider range of 2CL operators.

Constitutive Specification of RONR.

- (a) $motion(c, m)$ **means** $\forall_{p_i \in P} CREATE(C(c, p_i, cfv(m)))$
- (b) $openDebate(c, p, m)$ **means** $CREATE(C(c, p, assignFloor(p, m)))$
- (c) $refuseFloor(p, c, m)$ **means** $refusedFloor(p, m) \wedge$
 $RELEASE(C(c, p, assignFloor(p, m)))$
- (d) $askFloor(p, c, m)$ **means** $CREATE(C(p, c, discussed(p, m)))$
- (e) $recognition(c, p, m)$ **means** $assignFloor(p, m)$
- (f) $startTalk(p, m)$ **means** $discussing(p, m)$
- (g) $stopTalk(p, m)$ **means** $discussed(p, m) \wedge \neg discussing(p, m) \wedge \neg assignFloor(p, m)$
- (h) $timeOut(c, p)$ **means** $discussed(p, m) \wedge \neg discussing(p, m) \wedge$
 $RELEASE(C(p, c, discussed(p, m))) \wedge \neg assignFloor(p, m)$
- (i) $cfv(c, m)$ **means** $cfv(m)$
- (l) $vote(p, m)$ **means** $voted(p, m)$

All participants must be guaranteed to have the possibility to vote the presented motion. To this aim, by action *motion*, the chair commits to each participant to let him/her vote the motion, (a). These commitments will be discharged when the chair calls for votes, (i). The other actions are used in the discussion phase. For instance, the action *askFloor* is performed by one of the participants, who commits to discuss the motion, (d). This commitment is discharged by *stopTalk*, (g), which means that the participant has finished to speak and, for this reason, it loses the floor. The other actions are quite straightforward. Notice that, standing to the constitutive specification only, actions could be executed in *any order*, so the constitutive specification is not sufficient to guarantee that all participants will have the possibility to discuss the presented motion and vote.

Regulative Specification of RONR.

The regulative rules of RONR can be specified in 2CL as follows:

- c1: $C(c, p, cfv(m)) \bullet \rightarrow C(c, p, assignFloor(p, m))$
- c2: $C(c, p, assignFloor(p, m)) \bullet \rightarrow$
 $C(p, c, discuss(p, m)) \text{ XOR } refusedFloor(p, m)$
- c3: $C(p, c, discuss(p, m)) \bullet \rightarrow assignFloor(p, m)$
- c4: $assignFloor(p, m) \Rightarrow discussed(p, m)$
- c5: $assignFloor(p, m) \bowtie - discussing(p, m)$
- c6: $discussing(p, m) \bullet \rightarrow discussed(p, m)$

- $c7: \text{discussed}(p, m) \bullet \rightarrow \neg \text{assignFloor}(p, m)$
 $c8: \text{refusedFloor}(p, m) \text{ OR } \text{discussed}(p, m) \bullet \rightarrow \text{cfv}(m)$
 $c9: \text{cfv}(m) \rightarrow \text{voted}(p, m)$
 $c10: \text{assignFloor}(X, _) \not\equiv \text{assignFloor}(Y, _) \text{ AND } X \neq Y$

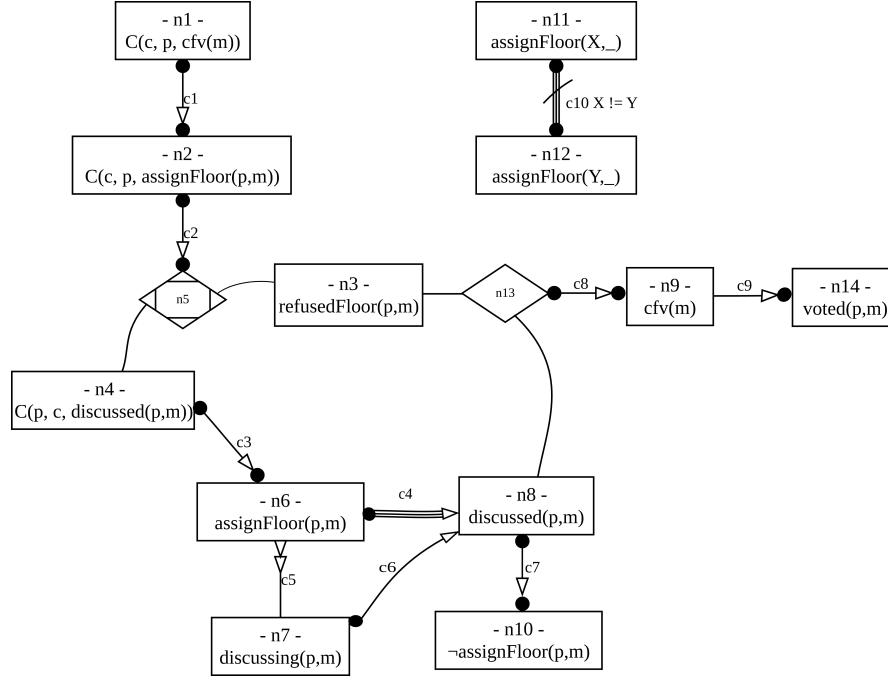


Fig. 5. Regulative specification of RONR.

$c1$, $c2$, $c3$ (as well as, further below, $c7$ and $c8$) are again examples of use of the *causes* operator ($\bullet \rightarrow$): $c1$ states that if the chair committed to allow a participant to vote, then it must also commit to give that participant the floor; $c2$ that if the social state contains a commitment to assign the floor to a participant, that participant can alternatively commit to discuss or refuse to speak; $c3$ that if a participant committed to discuss a motion, sooner or later it will be given the floor. Instead, $c4$ ($\bullet \Rightarrow$) is an operator of type *persistence* (persistence response): it states that when the floor is given to a participant, that participant *will keep it until* it will finish to speak. Constraint $c5$ ($\bullet \rightarrow$) involves two subsequent states, saying that the assignment of the floor is a *premise* to the discussion. Constraint $c6$ uses the *response* operator ($\bullet \rightarrow$): when a participant starts to discuss a motion, sooner or later it has to finish. Constraint $c7$ ($\bullet \rightarrow$) states that once a participant has finished to speak, then it will lose the floor; $c8$ (of the same kind) states that votes are to be called for only after each participant has either refused to speak or has already spoken. Let us recall, in fact, that the action *motion* creates a commitment of kind $C(c, p, \text{cfv}(m))$ for every participant. This is the reason why the call for votes can be done only after all of them have either spoken or refused to do it. Constraint $c9$ (\rightarrow) uses the *before* operator and states that voting can be done only after the proper call. Last but not the least constraint $c10$ ($\not\equiv$) is an example of use of a *negative operator*: it states that the floor cannot be assigned to two participants at the same time. Figure 5 reports the same constraints as a graph.

RONR shows how in certain contexts the regulative specification of an interaction protocol is not just a guideline because it is fundamental in order to give guarantees to the participants. On the other hand, the protocol is flexible because, for instance, it does not specify a predefined number of rounds of discussion. A participant may raise the hand as many times as he/she wishes for obtaining the floor. This flexibility is maintained by the 2CL representation. In fact, the constraints do not specify a flow of action execution but rather, following the motto *no flow in flow*, they rule the evolution of the social state in a declarative way. Moreover, the language allows to easily modify the protocol so to adapt it to different needs. For instance, by constraint *c9* participants can abstain from voting. If one changes the constraint to $\bullet \rightarrow$ then every participant would be expected to express his/her vote.

4. TAILORING THE CONTRACT NET PROTOCOL

Let us show the versatility of the proposed representation by showing how easy it is to vary a given protocol, by playing with its regulative specification separately from the constitutive specification of its actions. We show this by producing variations of the CNP, which suit different needs. All the variations are produced by working exclusively on the regulative specification without modifying the constitutive specification. Of course, it is possible to do the opposite or to modify both parts if needed.

Lazy and zealous participant

The first two simple variants are obtained by changing a single arrow with another operator from Table II. With reference to Fig. 6(a), if we substitute the *cause* relation in *c1* with a *before* relation (\rightarrow), the participant *will not be obliged* to answer with a proposal to the call of the initiator (it is allowed to have an unresponsive behavior). In constraints, the whole variant is:

- c1: $C(i, p, assigned_task(i, p)) \rightarrow C(p, i, solved_task(p, i)) \text{ XOR } refused_task(p, i)$
- c2: $C(p, i, solved_task(p, i)) \bullet \rightarrow rejected_proposal(i, p) \text{ XOR } assigned_task(i, p)$
- c3: $assigned_task(i, p) \bullet \rightarrow solved_task(p, i) \text{ XOR } failed(p, i)$

Instead, Fig. 6(b), if one uses in *c1* a *response* ($\bullet \rightarrow$) operator, the participant *can also take the initiative* to solve a task even though the initiator has not made any request (zealous participant), as it happens in the case of advertisement, where a provider promotes its services:

- c1: $C(i, p, assigned_task(i, p)) \bullet \rightarrow C(p, i, solved_task(p, i)) \text{ XOR } refused_task(p, i)$
- c2: $C(p, i, solved_task(p, i)) \bullet \rightarrow rejected_proposal(i, p) \text{ XOR } assigned_task(i, p)$
- c3: $assigned_task(i, p) \bullet \rightarrow solved_task(p, i) \text{ XOR } failed(p, i)$

These two variants correspond to protocols that differ from CNP but that can easily be obtained by working at the level of constraints among commitments, without modifying the constitutive specification.

CNP with Anticipated Failure

The next context that we consider is a call for bids, where an initiator publishes an open call, e.g. in an official gazette, that does not require the subscribers to the gazette to answer. Fig. 6(c) shows the new protocol: the fact that the participant is not obliged to send a bid is captured by the constraint *c1*, which is a *before* (\rightarrow) instead of being a *cause* ($\bullet \rightarrow$, as in Fig. 4). We have further modified the CNP by changing the constraint *c3* in Fig. 4, in order to capture the fact that a participant can notify a failure in the task solution also in the case in which the task has not been assigned to it yet but, for some reason, it has found out that it has become impossible for it to proceed with the solution, in case the task is assigned to it. Instead, it is not allowed to communicate the

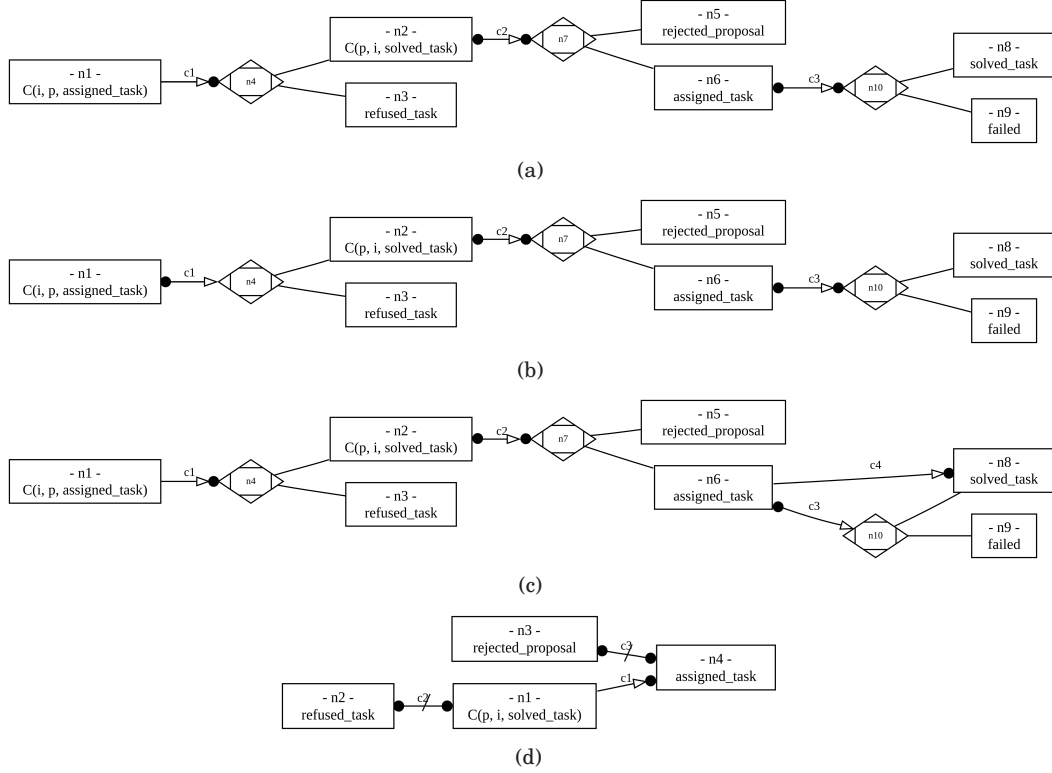


Fig. 6. (a) Lazy Participant; (b) Zealous Participant; (c) CNP with Anticipated Failure; (d) Soft CNP.

solution until the task is assigned to it. The new protocol can be obtained by modifying the regulative specification as in Fig. 6(c). In rules:

- c1:** $C(i, p, assigned_task(i, p)) \rightarrow C(p, i, solved_task(p, i)) \text{ XOR } refused_task(p, i)$
c2: $C(p, i, solved_task(p, i)) \bullet \rightarrow rejected_proposal(i, p) \text{ XOR } assigned_task(i, p)$
c3: $assigned_task(i, p) \bullet \rightarrow solved_task(p, i) \text{ XOR } failed(p, i)$
c4: $assigned_task(i, p) \rightarrow solved_task(p, i)$

The changes concern the constraints after node $n6$. In the new version, instead of having simply a *cause* constraint, we have a *response* ($\bullet \rightarrow$). Response is a softer constraint because it does not forbid to the alternatives specified by $n10$ to hold before $assigned_task(i, p)$. For this reason, in order to enforce that the solution is communicated only after the assignment, another constraint is to be added ($c4$). In this way, failure can be notified at any moment.

Soft CNP

The last example (Fig. 6(d)) is a very soft interaction protocol that, differently than the previous ones, expresses just a few regulative constraints, leaving a much greater freedom of behavior to the initiator and to the participant.

- c1:** $C(p, i, solved_task(p, i)) \rightarrow assigned_task(i, p)$
c2: $refused_task(p, i) \not\bullet C(p, i, solved_task(p, i))$
c3: $rejected_proposal(i, p) \not\bullet assigned_task(i, p)$

This example also shows the use of *negative* constraints. The only constraint that is imposed on the evolution of the social state is that a task cannot be assigned to a participant who has not yet committed to solve it. Moreover, there are two negative constraints (of kind \neg) stating that proposal assignment and rejection are mutually exclusive ($c3$), and that the refusal of a task is mutually exclusive to the commitment to solve it ($c2$). So, for instance, a participant can express its intention to solve a task for which no call has been made, and it is also possible for it to give a solution before any assignment of the task has been made to it. On the other hand, the initiator can ignore the participant even though it has committed to solve the task by avoiding to answer to it. It can call for proposals even if it already has a commitment by the participant, and it can reject a participant even though it has not made any proposal. It is not even necessary that the initiator commits to assign the task.

5. DISCUSSION ABOUT COMMITMENTS AND CONSTRAINTS

After the example and before the concluding remarks, this section discusses some relevant issues concerning the relation between commitments and constraints, including some considerations on the autonomy of the agents.

Can commitments substitute constraints?

For what concerns whether constraints can be represented (and, then, substituted) by proper commitments, this is not possible because the expressive power of 2CL constraints and of commitments is different. Let us show this with the help of the CNP, used in Section 4, Fig. 4. Here there is the need to express a temporal ordering between the assignment of a task to a participant and the fact that the participant has solved (or failed to solve) the task: the former condition is to be achieved first. The only way in which it is possible to express relations between conditions by means of commitments is by using a commitment of the kind:

$$C(p, i, \text{assigned_task}(i, p), \text{solved_task}(p, i) \text{ xor } \text{failed}(p, i))$$

By this commitment the participant p commits to the initiator i either to do perform a given task or to declare that the task execution failed. Here *xor* denotes the logical connective. This is to be done conditioned to the fact that the initiator i assigned p the task at issue ($\text{assigned_task}(i, p)$). Different evolutions of the interaction are allowed. In particular, if p happens to have solved the task before it is assigned to it, it can add to the social state $\text{solved_task}(p, i)$ and the commitment is discharged. This means that the antecedent conditions (like $\text{assigned_task}(i, p)$) that are used inside commitments are not necessarily to be accomplished before the engagement to bring about the consequent conditions (like $\text{solved_task}(p, i) \text{ xor } \text{failed}(p, i)$). For this reason commitments cannot be used to impose temporal orderings. Instead, constraints are specifically thought to express temporal relations. In the above case one could, for instance, specify (see also $c1$ in Section 4):

$$\text{assigned_task}(i, p) \bullet \rightarrow \text{solved_task}(p, i) \text{ XOR } \text{failed}(p, i)$$

meaning that the two conditions are to be achieved sequentially.

Are there differences in the nature of engagements?

Another important issue related to the (dis)similarities between constraints and commitments concerns the *nature of the engagements* that they represent. Ever since their introduction, commitments have been given an explicit normative nature [Castelfranchi 1995; Singh 1999]. Also constraints, which define the patterns of interaction of a protocol, have a normative nature, in the sense that they represent what must hold in an execution for a protocol to be respected. Both constraints and commitments, due

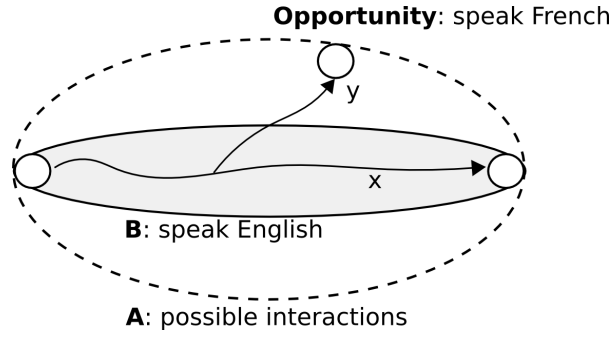


Fig. 7. *A*: space of possible social state evolutions according to the constitutive specification of actions; *B*: subset of evolutions that respect the regulative specification of the interaction protocol; *x*: specific interaction that leads to a final state containing no commitments, that is foreseen by the protocol; *y*: interaction where, at some point, agents leave the protocol reaching a final state that may even contain no commitment.

to their regulative nature, introduce a notion of *violation*. There are, however, some differences. An agent takes a commitment by autonomously deciding to execute an action, whose public and shared meaning includes that commitment. Afterwards, it is bound to make the condition in the commitment become true. As soon as the condition of the commitment is achieved, the commitment is discharged. The debtor paid its debt. On the other hand, an agent that accepts to interact according to a protocol accepts to stick to the patterns of interaction that the protocol specifies. The agent is free to execute any action, as long as it respects the rules. While a commitment requires the debtor to satisfy it sooner or later, a constraint is to be respected for the whole interaction. A consequence of the different nature of commitments and of constraints concerns the *time* at which *violations can be detected*. Violations of commitments can be detected only at the end of the interaction: an ending state containing a commitment means that the debtor did not comply to what expected from it. Violations to constraints, instead, can be detected during the interaction. A constraint is, in fact, like a boundary that should not be crossed.

So far, 2CL statements have not been given an explicit deontic semantics because in this stage of the work the focus was the identification of the most relevant kinds of constraints together with their intuitive semantics and with a formal characterization of it. We are currently working at a deontic characterization of the proposal that is based on *dialectical commitments* [Singh 2008]. These are similar in nature to the practical commitments used in this proposal because they have a regulative nature. However, they allow to express *maintenance*, rather than achievement, engagements as well as constraints do.

What about profiting of opportunities?

The classical commitment-based approach is very respectful of the agents' autonomy. Autonomy implies that each agent decides what is the best for itself. For this reason, protocols do not dictate agents when to execute specific actions. In particular, an agent can decide to take an opportunity, when one arises. This spirit is respected by our proposal, even though our protocols include a regulative specification which restricts the acceptable executions, because we do not regiment the regulative rules. An agent can always decide to break the rules if a situation it can take advantage of arises, although this amounts to a violation in our framework. Let us explain the meaning of opportunity, and the possibility to incur into a violation, with the help of a simple example: suppose that at a summer school the official language everybody should speak is En-

lish. Every student at the school has a badge reporting the name and the nationality of the person. If a French attendee meets a colleague whose badge says she is from France, the first student might decide to speak in French with the other even though the official language she should speak is English. There is a clear expected advantage in doing so: a better understanding (see Figure 7). However, the violation of the rules of the protocol introduces a certain amount of *risk*. For example, if the second student by mistake took the badge of her roommate, she might not understand the former one because the protocol is not attended. This would not have happened if the first agent did not violate the rules.

Do one's commitments commit others to commit?

A final consideration concerns constraints that relate commitments with different debtors, like for instance:

$$C(c, m, \text{purchase}(\text{goods})) \bullet \leftrightarrow C(m, c, \text{sold}(\text{goods}, \text{price}))$$

by which the commitment of the customer c to buy some goods ($C(c, m, \text{purchase}(\text{goods}))$) is to be followed by the commitment of the merchant m to sell the goods at some agreed price ($C(m, c, \text{sold}(\text{goods}, \text{price}))$). It may seem, in this case, that the autonomy of the merchant is reduced by the simple fact that another agent took some commitment, but it is not so. In fact, since constraints are supposed to be public and inspectable specifications, an agent, willing to play a role in a protocol, has the means for understanding if that pattern of interaction meets its goals. By autonomously deciding whether entering the protocol it, however, *commits* to respect its rules along the *whole* interaction. The fact that all agents accept to respect the rules has the advantage of making the course of interaction predictable and, therefore, of giving guarantees to all of the participants. In the example, the customer has the guarantee that the merchant will take the commitment to sell the goods after it commits to buy.

6. CONCLUSIONS AND FUTURE WORK

Constitutive and regulative specifications have been recognized as fundamental components of the interaction based on communication starting from Searle [Searle 1969; Cherry 1973; Searle 1995], and including various authors in the Multi-Agent community, e.g. [Chopra and Singh 2008; Chopra 2009; Boella and van der Torre 2004; Grossi 2007]. In this work we have presented a model of commitment-based interaction protocols that includes an explicit representation of both constitutive and regulative specifications, and we have discussed its operational semantics. In particular, the model adopts an operational semantics of commitments, given by means of commitment machines, and adds to classical commitment-based protocols an explicit representation of articulated regulative specifications, expressed by means of constraints among commitments. In order to specify such constraints, we have proposed the language 2CL, which, from a graphical point of view, is inspired to [Pesic and van der Aalst 2006; Baldoni et al. 2011]. The operators of 2CL have been formalized by means of LTL.

Other approaches, like [Giordano et al. 2007; Singh 2008; Bentahar et al. 2009; El-Menshaway et al. 2010; 2011], model commitments as modal connectors and exploit semantic frameworks based on temporal logics. In particular, [El-Menshaway et al. 2010; 2011] propose a branching-time logic that extends CTL*, named CTL^{*sc} . The main advantage of this approach stands in the fact that it supplies a logical semantics for the operations, that are used to manipulate commitments. This approach was designed to perform verifications of agent interactions ruled by classical commitment-based protocols by exploiting symbolic model checking techniques. The properties that can be verified are inspired by the ones that are commonly checked in communications protocols for distributed systems: fairness, safety, liveness, and reachability. As observed

in Section 2, the regulative specification of the commitment-based protocols, used in these works, is limited to the use of commitments themselves. It would be interesting to integrate in this logical framework the 2CL constraints in order to combine the benefits of both approaches: on the one hand, the possibility to embed in the protocols expressive regulative specification, and, on the other hand, the possibility to exploit the logical framework to perform the listed verifications. Notice that this integration would be quite easy because the LTL meaning of 2CL constraints is preserved by a transformation in CTL*: it is sufficient to properly apply the universal path quantifier A to the formulas.

An interesting connection can be observed between our proposal and the one in [Marengo et al. 2011], where temporal regulations, that capture a workflow of execution (defined on top of events), can be used as the antecedent or consequent conditions of commitments. This allows the reification of regulations into business relationships. The key difference is that while in our proposal constraints can be interpreted as maintenance engagements, in [Marengo et al. 2011] workflows inside commitments amount to achievement engagements.

Commitment-based protocols, in general, can be seen as context-dependent in that they allow the specification of the social meaning of any kind of action, not necessarily of communicative actions and not necessarily of standardized actions. Our proposal extends commitment-based protocols and for this reason it keeps the same generality. This characteristic is due to the fact that such protocols are thought as a “standard to define standards”. It does not prevent the use of standardized actions, when desired, but does not preclude the definition of other actions. Indeed, we agree with the claim “Communication is inherently sensitive to its social or organizational context. But context-sensitivity is at odds with standardization: how may we reconcile the two? I claim there cannot be a globally valid ‘standard’ definition of agent communication (AC) primitives. Researchers should give up looking for such definitions. Instead, I advocate a methodology by which a community of practice would create and maintain its own dialect.” [Singh 2010]. The adaptation of commitment-based protocols to different contexts of use has been tackled in [Chopra and Singh 2006]. The authors show how declarative approaches suit well this aim. Our proposal is set along this line. In fact, not only the constitutive rules are given in a declarative way but also the regulative specification is made of declarative constraints and it is possible to contextualize it by adding or removing constraints. The advantage with respect to [Chopra and Singh 2006] is the modularity of the two specifications discussed along the paper.

For what concerns the future developments of this research, in [Baldoni et al. 2010] we proposed to use our approach to protocol specification to give semantics to *environments* [Omicini et al. 2008]. The aim is to develop infrastructures that explicitly represent not only the agents but also the computational environment in terms of rules, conventions, resources, tools, and services that are functional to the coordination and cooperation of the agents. The inspiration is the MAS architecture sketched in [Chopra and Singh 2009a], and the idea is to introduce also the possibility to detect violations as it is done in *Electronic Institutions*, like [Esteva et al. 2004; Arcos et al. 2007; Fornara et al. 2008], and in *normative artifacts*, like [Dastani et al. 2008]. This latter proposal, as well as ours, is specifically designed to rule the observable behavior of agents but, differently than ours, the representation of norms takes into account the current state of execution only and, therefore, cannot express temporal relations like the ones that we have described. Along this line, another interesting proposal is made in [Hübner et al. 2010], where an organization is compiled into a normative system. Also this proposal is based on the use of computational environments, and exploits regimentation to implement the norms.

The new framework will enable the verification of interaction properties of MAS from the global point of view of the system as well as from the point of view of the single agents along the lines of the proposal in [Baldoni et al. 2011]. Specifically, we foresee three basic kinds of verifications: monitoring if the current executions complies to the constraints (or verifying if a recorded execution complies to the constraints); enabling agents to check whether their own procedures conform to the specifications, and whether playing according to the protocol rules will allow them to reach their goals; verifying properties of the interaction protocol itself. To these aims different techniques and tools could be used, ranging from model checking to theorem proving and to colored transition systems, as proposed in [Sergot 2007]. For what concerns the third kind of verification, we can also envisage to study the possible uses of *Normative Temporal Logic* (NTL) [Ågotnes et al. 2007], a generalization of the branching-time temporal logic (CTL), where path quantifiers A and E are replaced by the deontic operators O_η and P_η respectively (the obligation and permission operators are applied to the normative system η). NTL could, in fact, help to prove properties of a model of interaction as expressed in our proposal, where the set of possible executions is restricted according to specific regulations.

An approach similar to commitment-based protocols is the one introduced by Alberti et al. [2004], where *expectation*-based protocols are presented. Expectations concern events expected to happen (or not to happen) and can be associated to time points. Protocols are specified by constraining the times at which events occur. As for the previous works, the limit of this approach is that it works directly on events (i.e. actions); by constraining actions the approach lacks the openness discussed in the Introduction and in the discussion about ConDec in Section 2. On the other hand, our proposal does not handle time explicitly so we cannot yet represent and handle timeouts and also compensation mechanisms. The aim of this paper is, indeed, to present the idea of an explicit, declarative, and decoupled representation of both the constitutive and the regulative specifications. We mean to tackle also issues concerning time, deadlines, faults and compensation, like in [Torrioni et al. 2010] (where commitments are implemented by means of expectations), in future work.

The work by Miller and McGinnis [2008] contains a comparison of various approaches to interaction protocols, including but not limited to commitment-based protocols. Specifically, also normative systems, algebraic and procedural approaches (like *RASA* [Miller and McBurney 2008] and the seminal work in [McBurney and Parsons 2002]), and Petri nets are considered. The comparison is done along many directions. The authors confirm our opinion that declarative approaches (like commitment-based ones) are very flexible. However, they claim that they are less readable (and sometimes more verbose) than algorithmic approaches. To support this consideration they cite some of the major existing tools for the designer (like AgentUML [FIPA Modeling TC 2007]), which are algorithmic. For verbosity, they cite the CNP representation in [Alberti et al. 2004] which consists of seventeen rules. We underline that our regulative representation of CNP consists instead of *three* rules only. The constitutive specification is made of *seven* rules (because there are seven actions). For what concerns commitment protocols, the difficulty in reading declarative specifications is, in our opinion, due to the lack of separation between the constitutive and the regulative specifications that many approaches show. Moreover, as van der Aalst et al. [2005] notice, there is a lack of graphical intuitive representations oriented to designers. We have tried to overcome these problems by decoupling the regulative and the constitutive specifications and by giving a graphical representation. This representation has the advantage of giving the *perception of a flow* in the execution, remaining however at a *what* rather than at a *how* level (we call this *no-flow in-flow*). This representation

also supports the compositionality of the protocols. In fact, to put it simply, in order to produce a new protocol starting from existing ones, it is sufficient to draw together the sets of constraints of interest and produce a bigger graph without any effort. Protocols can, then, be designed bottom-up. We mean to study, as future work, a methodology that allows the achievement of compositionality in declarative protocols. The intuition is that the decoupling of the regulative and constitutive specifications will facilitate the specification of a methodology.

ACKNOWLEDGMENTS

The authors thank Guido Boella for the interesting discussions. The authors would like to thank the anonymous reviewers for their comments and suggestions, which helped to improve the paper.

REFERENCES

- ÅGOTNES, T., VAN DER HOEK, W., RODRÍGUEZ-AGUILAR, J. A., SIERRA, C., AND WOOLDRIDGE, M. 2007. On the Logic of Normative Systems. In *Proc. of International Joint Conference on Artificial Intelligence (IJCAI 2007)*, M. M. Veloso, Ed. AAAI Press, Hyderabad, India, 1175–1180.
- ALBERTI, M., DAOLIO, D., TORRONI, P., GAVANELLI, M., LAMMA, E., AND MELLO, P. 2004. Specification and Verification of Agent Interaction Protocols in a Logic-based System. In *Proc. of the 2004 ACM Symposium on Applied Computing (SAC 2004)*, H. Haddad, A. Omicini, R. L. Wainwright, and L. M. Liebrock, Eds. ACM, Nicosia, Cyprus, 72–78.
- ARCOS, J. L., NORIEGA, P., RODRÍGUEZ-AGUILAR, J. A., AND SIERRA, C. 2007. E4MAS Through Electronic Institutions. In *Environments for Multi-Agent Systems III, Third International Workshop (E4MAS 2006)*, D. Weyns, H. Van Dyke Parunak, and F. Michel, Eds. LNCS Series, vol. 4389. Springer, Hakodate, Japan, 184–202.
- ARTIKIS, A., O'HARE, G. M. P., STATHIS, K., AND VOUIROS, G. A., Eds. 2008. *Proc. of the 8th International Workshop on Engineering Societies in the Agents World VIII (ESAW 2007)*. LNCS Series, vol. 4995. Springer, Athens, Greece.
- BALDONI, M., BAROGLIO, C., BERGENTI, F., BOCCALATTE, A., MARENGO, E., MARTELLI, M., MASCARDI, V., PADOVANI, L., PATTI, V., RICCI, A., ROSSI, G., AND SANTI, A. 2010. MERCURIO: An Interaction-oriented Framework for Designing, Verifying and Programming Multi-Agent Systems. In *Proc. of 11th International Workshop on Coordination, Organization, Institutions and Norms in Multi-Agent Systems (COIN@MALLOW 2010)*, N. Fornara and G. Vouros, Eds. CEUR Workshop Proceedings, Lyon, France.
- BALDONI, M., BAROGLIO, C., BRUNKHORST, I., HENZE, N., MARENGO, E., AND PATTI, V. 2011. Constraint Modeling for Curriculum Planning and Validation. *International Journal of Interactive Learning Environments* 19, 1, 83–123.
- BALDONI, M., BAROGLIO, C., AND MARENGO, E. 2010. Behavior-Oriented Commitment-Based Protocols. In *Proc. of the European Conference on Artificial Intelligence (ECAI 2010)*, H. Coelho, R. Studer, and M. Wooldridge, Eds. Frontiers in Artificial Intelligence and Applications Series, vol. 215. IOS Press, Lisbon, Portugal, 137–142.
- BALDONI, M., BAROGLIO, C., AND MARENGO, E. 2011. Commitment-based Protocols with Behavioral Rules and Correctness Properties of MAS. In *Post-Proc. of the International Workshop on Declarative Agent Languages and Technologies (DALT 2010)*, A. Omicini, S. Sardina, and W. Vasconcelos, Eds. LNAI. Springer, Toronto, Canada.
- BALDONI, M., BENTAHAR, J., VAN RIEMSDIJK, M. B., AND LLOYD, J., Eds. 2010. *Proc. of the 7th International Workshop on Declarative Agent Languages and Technologies VII (DALT 2009)*. LNCS Series, vol. 5948. Springer, Budapest, Hungary.
- BENTAHAR, J., MEYER, J.-J. C., AND WAN, W. 2009. Model Checking Communicative Agent-based Systems. *Knowledge-Based Systems* 22, 3, 142–159.
- BOELLA, G. AND VAN DER TORRE, L. W. N. 2004. Regulative and Constitutive Norms in Normative Multi-agent Systems. In *Proc. of the Ninth International Conference on Principles of Knowledge Representation and Reasoning (KR2004)*, D. Dubois, C. A. Welty, and M.-A. Williams, Eds. AAAI Press, Whistler, Canada, 255–266.
- CASTELFRANCHI, C. 1995. Commitments: From Individual Intentions to Groups and Organizations. In *Proc. of the First International Conference on Multiagent Systems (ICMAS 1995)*, V. R. Lesser and L. Gasser, Eds. The MIT Press, San Francisco, California, USA, 41–48.
- CHERRY, C. 1973. Regulative Rules and Constitutive Rules. *The Philosophical Quarterly* 23, 93, 301–315.

- CHESANI, F., MELLO, P., MONTALI, M., AND TORRONI, P. 2009. Verifying A-Priori the Composition of Declarative Specified Services. In *Proc. of the Second Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW 2009)*, M. Baldoni, C. Baroglio, J. Bentahar, G. Boella, M. Cossentino, M. Dastani, B. Dunin-Keplicz, G. Fortino, M. P. Gleizes, J. Leite, V. Mascardi, J. A. Padget, J. Pavón, A. Polleres, A. El Fallah-Seghrouchni, P. Torroni, and R. Verbrugge, Eds. CEUR Workshop Proceedings Series, vol. 494. CEUR-WS.org, Turin, Italy.
- CHOPRA, A. 2009. Commitment Alignment: Semantics, Patterns, and Decision Procedures for Distributed Computing. Ph.D. thesis, North Carolina State University, Raleigh, NC.
- CHOPRA, A. K. AND SINGH, M. P. 2006. Contextualizing Commitment Protocol. In *Proc. of 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, H. Nakashima, M. P. Wellman, G. Weiss, and P. Stone, Eds. ACM, Hakodate, Japan, 1345–1352.
- CHOPRA, A. K. AND SINGH, M. P. 2008. Constitutive Interoperability. See Padgham et al. [2008], 797–804.
- CHOPRA, A. K. AND SINGH, M. P. 2009a. An Architecture for Multiagent Systems: An Approach Based on Commitments. In *Proc. of the AAMAS Workshop on Programming Multiagent Systems (ProMAS 2009)*, L. Braubach, J.-P. Briot, and J. Thangarajah, Eds. LNAI Series, vol. 5919. Springer, Heidelberg, 148–162.
- CHOPRA, A. K. AND SINGH, M. P. 2009b. Multiagent Commitment Alignment. In *Proc. of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, C. Sierra, C. Castelfranchi, K. S. Decker, and J. S. Sichman, Eds. Vol. 2. IFAAMAS, Budapest, Hungary, 937–944.
- CLARKE, O. E. M. AND PELED, D. 2001. *Model Checking*. MIT Press, Cambridge, MA, USA.
- CONTE, A. G. 1983. *Regola costitutiva, condizione antinomia*. Edizioni di Comunità, Milano.
- DASTANI, M., GROSSI, D., MEYER, J. J. C., AND TINNEMEIER, N. A. M. 2008. Normative Multi-agent Programs and Their Logics. In *Knowledge Representation for Agents and Multi-Agent Systems, First International Workshop (KRAMAS 2008)*, J.-J. C. Meyer and J. Broersen, Eds. LNCS Series, vol. 5605. Springer, Sydney, Australia, 16–31.
- EL-MENSHAWY, M., BENTAHAR, J., AND DSSOULI, R. 2010. Verifiable Semantic Model for Agent Interactions Using Social Commitments. In *Languages, methodologies and Development tools for multi-agent systems (LADS 2009)*, M. Dastani, A. Seghrouchni El Fallah, J. Leite, and P. Torroni, Eds. LNCS Series, vol. 6039. Springer, Torino, Italy, 128–152.
- EL-MENSHAWY, M., BENTAHAR, J., AND DSSOULI, R. 2011. Symbolic Model Checking Commitment Protocols using Reduction. In *Declarative Agent Languages and Technologies VIII*, A. Omicini, S. Sardina, and W. Vasconcelos, Eds. Springer, Toronto, Canada.
- EMERSON, E. A. 1990. *Temporal and Modal Logic*. Vol. B. Elsevier, Amsterdam, The Netherlands.
- ESTEVA, M., ROSELL, B., RODRÍGUEZ-AGUILAR, J. A., AND ARCOS, J. L. 2004. AMELI: An Agent-Based Middleware for Electronic Institutions. In *Proc. of 3rd Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*. IEEE Computer Society, New York, NY, USA, 236–243.
- FIPA MODELING TC. 2007. FIPA Agent UML. <http://www.auml.org/>.
- FORNARA, N. 2003. Interaction and Communication among Autonomous Agents in Multiagent Systems. Ph.D. thesis, Università della Svizzera italiana, Facoltà di Scienze della Comunicazione.
- FORNARA, N. AND COLOMBETTI, M. 2003. Defining Interaction Protocols using a Commitment-based Agent Communication Language. See Rosenschein et al. [2003], 520–527.
- FORNARA, N. AND COLOMBETTI, M. 2004a. A Commitment-Based Approach To Agent Communication. *Applied Artificial Intelligence* 18, 9-10, 853–866.
- FORNARA, N. AND COLOMBETTI, M. 2004b. Protocol Specification Using a Commitment Based ACL. In *Advances in Agent Communication, International Workshop on Agent Communication Languages (ACL 2003)*, F. Dignum, Ed. LNCS Series, vol. 2922. Springer, Melbourne, Australia, 108–127.
- FORNARA, N., VIGANÒ, F., VERDICCHIO, M., AND COLOMBETTI, M. 2008. Artificial institutions: a model of institutional reality for open multiagent systems. *Artificial Intelligence and Law* 16, 1, 89–105.
- Foundation for Intelligent Physical Agents 2002. *FIPA Contract Net Interaction Protocol Specification*. Foundation for Intelligent Physical Agents.
- GIORDANO, L., MARTELLI, A., AND SCHWIND, C. 2007. Specifying and Verifying Interaction Protocols in a Temporal Action Logic. *Journal of Applied Logic* 5, 2, 214–234.
- GROSSI, D. 2007. Designing Invisible Handcuffs, Formal Investigations in Institutions and Organizations for Multi-agent Systems. Ph.D. thesis, University of Utrecht.
- GROSSI, D., ALDEWERELD, H., AND DIGNUM, F. 2007. Ubi lex, ibi poena: Designing norm enforcement in e-institutions. In *Proc. of Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems II (COIN 2006)*, V. Dignum, N. Fornara, P. Noriega, G. Boella, O. Boissier, E. Matson, and J. Vázquez-Salceda, Eds. LNCS Series, vol. 4386. Springer, Hakodate, Japan, 101–114.

- HÜBNER, J. F., BOISSIER, O., AND BORDINI, R. H. 2010. From Organisation Specification to Normative Programming in Multi-Agent Organisations. In *Proc. of the 11th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA XI)*, J. Dix, J. Leite, G. Governatori, and W. Jamroga, Eds. LNCS Series, vol. 6245. Springer, Lisbon, Portugal, 117–134.
- JONES, A. J. I. AND SERGOT, M. 1994. *On the Characterization of Law and Computer Systems: the Normative Systems Perspective*. John Wiley & Sons, Inc., New York, NY, USA, 275–307.
- JONES, A. J. I. AND SERGOT, M. J. 1996. A Formal Characterisation of Institutionalised Power. *Logic Journal of the IGPL* 4, 3, 429–445.
- KAFALI, Ö. AND YOLUM, P. 2009. Detecting Exceptions in Commitment Protocols: Discovering Hidden States. In *Proc. of the Second Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW 2009)*. CEUR Workshop Proceedings Series, vol. 494. CEUR-WS.org, Turin, Italy.
- LA TORRE, M. 2010. *Law as Institution*. Law and Philosophy Library Series, vol. 90. Springer, Heidelberg.
- LAMPART, L. 1978. Time, Clocks, and the Ordering of Events in a Distributed System. *Communications of the ACM* 21, 7, 558–565.
- MALLYA, A. U. AND SINGH, M. P. 2006. Introducing Preferences into Commitment Protocols. In *Agent Communication II, International Workshops on Agent Communication (AC 2005 and AC 2006)*, F. Dignum, R. M. van Eijk, and R. A. Flores, Eds. LNCS Series, vol. 3859. Springer, Utrecht, Netherlands, 136–149.
- MARENGO, E., BALDONI, M., BAROGLIO, C., CHOPRA, A. K., PATTI, V., AND SINGH, M. P. 2011. Commitments with Regulations: Reasoning about Safety and Control in REGULA. In *Proc. of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*. IFAAMAS, Taipei, Taiwan.
- MCBURNEY, P. AND PARSONS, S. 2002. Games That Agents Play: A Formal Framework for Dialogues between Autonomous Agents. *Journal of Logic, Language and Information* 11, 3, 315–334.
- MILLER, T. AND MCBURNEY, P. 2008. Annotation and Matching of First-class Agent Interaction Protocols. See Padgham et al. [2008], 805–812.
- MILLER, T. AND MCGINNIS, J. 2008. Amongst First-Class Protocols. See Artikis et al. [2008], 208–223.
- MONTALI, M. 2010. *Specification and Verification of Declarative Open Interaction Models: a Logic-Based Approach*. LNBIP Series, vol. 56. Springer, Heidelberg.
- MONTALI, M., PESIC, M., VAN DER AALST, W. P., CHESANI, F., MELLO, P., AND STORARI, S. 2010. Declarative Specification and Verification of Service Choreographies. *ACM Transactions on the Web* 4, 1, 3–62.
- OMICINI, A., RICCI, A., AND VIROLI, M. 2008. Artifacts in the A&A meta-model for multi-agent systems. *Journal of Autonomous Agents and Multi-Agent Systems* 17, 3, 432–456.
- PADGHAM, L., PARKES, D. C., MÜLLER, J., AND PARSONS, S., Eds. 2008. *Proc of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*. Vol. 2. IFAAMAS, Estoril, Portugal.
- PESIC, M., BOSNACKI, D., AND VAN DER AALST, W. M. P. 2010. Enacting Declarative Languages using LTL: Avoiding Errors and Improving Performance. In *Proc. of the 17th International SPIN Workshop on Model Checking of Software (SPIN 2010)*, J. van de Pol and M. Weber, Eds. LNCS Series, vol. 6349. Springer, Enschede, The Netherlands, 146–161.
- PESIC, M. AND VAN DER AALST, W. M. P. 2006. A Declarative Approach for Flexible Business Processes Management. In *Proc. of Business Process Management International Workshops (BPM 2006)*, J. Eder and S. Dustdar, Eds. LNCS Series, vol. 4103. Springer, Vienna, Austria, 169–180.
- PITT, J., KAMARA, L., SERGOT, M. J., AND ARTIKIS, A. 2006. Voting in Multi-Agent Systems. *The Computer Journal* 49, 2, 156–170.
- ROBERT, H. M. I., EVANS, W. J., HONEMANN, D. H., AND BALCH, T. J. 2000. *Robert's Rules of Order Newly Revised, 10th Edition*. Da Capo Press, Cambridge, Massachusetts.
- ROSENSCHEIN, J. S., SANDHOLM, T., WOOLDRIDGE, M., AND YOKOO, M., Eds. 2003. *Proc. of the Second International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2003)*. ACM, Melbourne, Australia.
- SEARLE, J. 1969. *Speech Acts*. Cambridge University Press, England.
- SEARLE, J. 1995. *The construction of social reality*. Free Press, New York.
- SERGOT, M. J. 2007. Action and Agency in Norm-Governed Multi-agent Systems. See Artikis et al. [2008], 1–54.
- SINGH, M. P. 1999. An Ontology for Commitments in Multiagent Systems. *Artificial Intelligence and Law* 7, 1, 97–113.
- SINGH, M. P. 2000. A Social Semantics for Agent Communication Languages. In *Issues in Agent Communication*, F. Dignum and M. Greaves, Eds. LNCS Series, vol. 1916. Springer, Heidelberg, 31–45.

- SINGH, M. P. 2003. Distributed Enactment of Multiagent Workflows: Temporal Logic for Web Service Composition. See Rosenschein et al. [2003], 907–914.
- SINGH, M. P. 2008. Semantical Considerations on Dialectical and Practical Commitments. In *Proc. of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008)*, D. Fox and C. P. Gomes, Eds. AAAI Press, Chicago, Illinois, USA, 176–181.
- SINGH, M. P. 2010. Community Standards for Agent Communication. Unpublished Draft, available at <http://www.csc.ncsu.edu/faculty/mpsingh/papers/drafts/Singh-AC-Manifesto.pdf>.
- SINGH, M. P. AND CHOPRA, A. K. 2010. Correctness Properties for Multiagent Systems. See Baldoni et al. [2010], 192–207.
- TORRONI, P., CHESANI, F., MELLO, P., AND MONTALI, M. 2010. Social Commitments in Time: Satisfied or Compensated. See Baldoni et al. [2010], 228–243.
- VAN DER AALST, W. M. P., DUMAS, M., TER HOFSTEDE, A. H. M., RUSSELL, N., VERBEEK, H. M. W., AND WOHEDE, P. 2005. Life After BPEL? In *Formal Techniques for Computer Systems and Business Processes, European Performance Engineering Workshop (EPEW 2005) and International Workshop on Web Services and Formal Methods (WS-FM 2005)*, M. Bravetti, L. Kloul, and G. Zavattaro, Eds. LNCS Series, vol. 3670. Springer, Versailles, France, 35–50.
- WINIKOFF, M., LIU, W., AND HARLAND, J. 2005. Enhancing Commitment Machines. In *Proc. of the Second International Workshop on Declarative Agent Languages and Technologies II (DALI 2004)*, J. A. Leite, A. Omicini, P. Torroni, and P. Yolum, Eds. LNCS Series, vol. 3476. Springer, New York, NY, USA, 198–220.
- YOLUM, P. AND SINGH, M. P. 2001. Designing and Executing Protocols Using the Event Calculus. In *Agents*. ACM, New York, NY, USA, 27–28.
- YOLUM, P. AND SINGH, M. P. 2002. Commitment Machines. In *Proc. of the 8th International Workshop on Intelligent Agents VIII (ATAL 2001)*, J.-J. C. Meyer and M. Tambe, Eds. LNCS Series, vol. 2333. Springer, Seattle, WA, USA, 235–247.

Received July 2010; revised January 2011; accepted March 2011